The Reconstruction Software for the Muon Ionization Cooling Experiment Trackers

The Reconstruction Software for the Muon Ionization Cooling Experiment Trackers

A Dobbs[1]**, K Long**[1]**, E Santos**[1]**, D Adey**[2]**, P Hanlet**[2]**, C Heidt**[3]

[1]Imperial College London, Blackett Laboratory, Prince Consort Road, London, SW7 2AZ, U.K.
[2]Fermilab National Accelerator Laboratory, Batavia, Illinois, U.S.A.
[3]University of California, Riverside, 900 University Ave, Riverside, CA 92521, U.S.A.

E-mail: adobbs@imperial.ac.uk

Abstract. The international Muon Ionisation Cooling Experiment (MICE) is designed to demonstrate the principle of muon ionization cooling, for application to a future Neutrino Factory or Muon Collider. In order to measure the change in emittance, MICE is equipped with a pair of high precision scintillating fibre trackers. The trackers are required to measure a 10% change in emittance to 1% accuracy (giving an overall precision of 0.1%).

This paper describes the tracker reconstruction software, as a part of the overall MICE software framework, MAUS. Channel clustering is described, proceeding to the formation of space-points, which are then associated with particle tracks using pattern recognition algorithms. Finally a full custom Kalman track fit is performed, to account for energy loss and multiple scattering. Exemplar results are shown for Monte Carlo data.

1. Introduction

1.1. The MICE Experiment

In order to produce high intensity muon beams, beam cooling, that is a reduction in beam phase space (or *emittance*), is necessary. Due to the short muon lifetime, traditional beam cooling techniques cannot be used; ionization cooling as a means of beam cooling was first proposed in the early 1970s (see Skrinsky [1]), but has yet to be demonstrated. Muon cooling is necessary for future facilities based on high intensity muon beams, such as a Neutrino Factory, the ultimate tool conceived for exploring leptonic CP violation, or a Muon Collider, a facility offering a potential route to multi-TeV lepton - anti-lepton collisions. MICE [2] is designed to demonstrate ionization cooling and, when finished, will consist of one complete cell of a Neutrino Factory cooling channel.

MICE is a staged experiment, that is, built and run in discrete sections. MICE is currently at Step I, that is a muon beamline with particle identification (PID) detectors, and has been taking data since the spring of 2008. The next stage of MICE, known as Step IV, is due to begin running in 2015. In Step IV, the two scintillating fibre trackers and a single absorber module of the cooling channel will be introduced. The final stage of MICE, Step VI, comprising the full cooling cell, is scheduled for 2019. A schematic of the full cooling channel is shown in Fig. 1.

Published under licence by IOP Publishing Ltd

1.2. The Trackers

The scintillating fibre trackers [3] are required to measure the emittance both before and after cooling. A schematic of a tracker is shown in Fig. 2. Each tracker is housed in a 4 T superconducting solenoid to produce helical charged particle tracks. There are five detector stations per tracker, set at varying distances along the direction of the beamline. Each station in turn consists of three fibre planes, the planes being orientated at 120° to each other. A plane is composed of 300 μm fibres, arranged in a doublet layer structure, seven fibres being ganged together to create single readout channel. Light produced by particles passing through the planes is extracted by waveguides and sent to visible light photon counters (VLPCs), to produce an electrical signal which is then digitised.

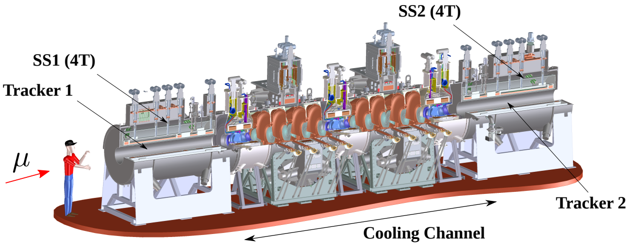

Figure 1. The full MICE Step VI cooling channel. “SS” refers to Spectrometer Solenoids, used to created helical particle paths for tracking.

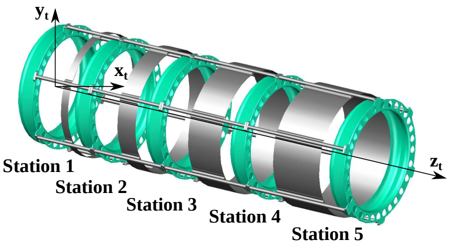

Figure 2. Schematic tracker frame, showing the station numbering and coordinate system.

This paper describes the software used to perform the reconstruction of data from the trackers, in particular the pattern recognition and Kalman filter track fit routines.

1.3. The MAUS Framework

The MICE Analysis User Software (MAUS) [4] framework performs the Monte Carlo (MC) simulation, reconstruction and analysis for MICE. It consists of a top level Python framework, which calls various lower level modules written in C++ or Python. The top level structure follows a functional coding style, consisting of four module types, namely Input, Mapper, Reducer and Output. MC simulation is provided by the GEANT4 [5] package, data persistency uses either the ROOT (binary) [6] or JSON (ascii) formats, and is analysis primarily via ROOT.

The tracker reconstruction software consists of three main mappers, which provide: digitisation, MC data digitisation and the subsequent reconstruction. There also exists a reducer module used to display reconstruction results to screen.

2. Data Structure

A simplified schematic of the tracker data structure, with the relevant entries from the more general MAUS data structure, is shown in Fig. 3.

The spill is the top level object, containing all the physics data for one ∼2 ms burst of particles through MICE. Below the spill level, the data structure is split into a MC and reconstruction side, the latter in principle being forbidden to know anything of the former, so that the reconstruction may proceed in an identical fashion for both real data and simulation. The only relevant element of the tracker data structure on the MC side is SF (“SciFi”, that is, scintillating fibre) Hits, which represent a single channel hit in a tracker plane. The MC digitisation process converts these hits to SF Digits (hereafter just called digits), with a similar process for real data also creating digits. The reconstruction side will be discussed in section 3 below.

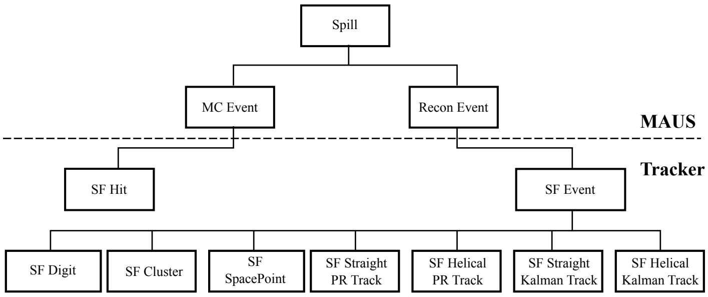

Figure 3. The tracker software data structure, and relevant MAUS data structure. The spill is the top level object below which data is split into MC and reconstruction sides.

3. Reconstruction

Digits are the most basic unit fed into the main reconstruction module, each digit representing a signal from one tracker channel. Digits from adjacent channels are assumed to come from the same particle and are grouped to form clusters. Clusters from channels which intersect each other, in at least two planes from the same station, are used to form space-points, giving x and y positions where a particle intersected a station. Once space-points have been found, they are associated with individual tracks through pattern recognition (PR) (described in section 4), giving straight or helical PR tracks. These tracks, and the space-points associated with them, are then sent to the final track fit (see section 5).

4. Pattern Recognition

4.1. Straight Tracks

In the absence of a magnetic field, the tracks passing through the tracker may be described using a straight line in three dimensions. Taking the z coordinate, as defined in Fig. 2, as the independent parameter, the track parameters are then the intercepts and gradients of the tracks in x and y, giving four parameters in all.

Tracks are formed when three or more stations have space-points present. Reconstruction proceeds by choosing one space point in each of two stations (ideally the first and last stations). Using the positions of the space-points a trial line is defined. A search is then made in the intermediate stations looking for space-points which come within a particular distance, known as the road, of this trial line. If at least one such space point is found, then they are all sent to a linear least-squares fitting algorithm, following the account in [7]. If this produces a track with an acceptably low χ^2, then the track is accepted and the space-points are marked as used. The procedure is repeated, looking for other tracks, until no more may be formed.

4.2. Helical Tracks

Helical tracks form when the solenoidal fields are active in the trackers. The tracks may be parameterised by first considering their projection in the x - y plane, which is a circle. The first three parameters are then radius, ρ and x, y coordinates of the circle centre, (X_0, Y_0). A coordinate ϕ may also be defined as the turning angle in this plane, in an anticlockwise sense, of each space point, which then allows the turning angle of most upstream space point to be the fourth parameter. Finally, defining the distance swept out by the circumference of the circle in the x - y plane as s, the helix forms a straight line in the s - z plane, allowing the gradient of this line, $\frac{\mathrm{d}s}{\mathrm{d}z}$, to be the final parameter (equivalent to the “dip angle” often used to define the “compactness” of a helix).

Reconstruction proceeds in a similar manner to before, starting in the x - y plane. Here, however, we do not form a trial circle of three space-points, and make road cuts, but rather move straight to a linear least squares fit after selecting space-points from each station. A linear least squares fit is still appropriate for a circle fit as, although the function in terms of position is non-linear, the parameters of the function, (ρ, X_0, Y_0), remain linear, and we again follow the procedure in [7]. If the χ^2 of the fit satisfies a minimum requirement, the space-points are accepted and we move on to fit in s - z.

Finding the true value of s is not as straight forward as it may first appear. The turning angles observed may have an extra $2n\pi$ rotations associated with them, if the track has undergone multiple rotations between stations. This can be resolved by exploiting the varying separation of the stations in z, and noting that the ratio of the difference of the turning angles between spacepoints in two stations divided by the difference in the separation in z, is a constant. Different values of n may then be tried, until one satisfying this criterion is found.

Once the number of extra rotations has been determined for each turning angle, we may calculate s simply using $s = \rho\phi$, then perform the straight line fit in s - z. If this fit passes a χ^2 cut, we accept these space-points as being part of the helical track and mark them as used. The gradient of the fit gives the required $\frac{\mathrm{d}s}{\mathrm{d}z}$ parameter. This is repeated until no more tracks may be formed. Examples of track visualisation for helical tracks are shown in Fig. 4.

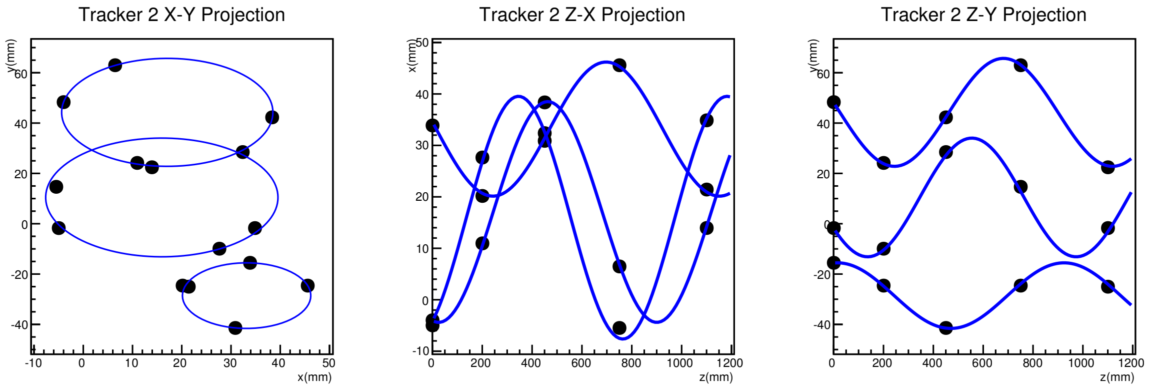

Figure 4. Example output of the visualisation modules for a spill containing three helical tracks, in the second tracker.

5. Final Track Fit with Kalman Filter

Once pattern recognition is complete, the tracks and their associated space-points and clusters are sent to the final track fit. The method used was introduced by Kalman in his 1960 seminal paper [8], and then became widespread in high energy physics after Fruhwirth's 1987 publication [9]. It is capable of producing high precision fits, accounting for both energy loss and multiple Coulomb scattering.

The general ingredients for Kalman track fitting are: a set of detector measurements grouped together as a track candidate (pattern recognition), a track model describing the path of the particle in the detector, the resolution of the detector measurements and, finally, the geometry of the detector itself and any material the particles may encounter. The latter are used to correct for the stochastic noise introduced by multiple Coulomb scattering and energy loss.

The incorporation of detector specifics in the Kalman routines is a delicate and important feature of this framework. In the current setup, the field is assumed to be uniform with magnitude 4 T. The cluster-level measurements are incorporated in the fit, rather than the higher level space point (x, y) values, as the space-points are a reconstructed data product and might introduce a bias. The success of the fit is evaluated by the goodness of the pull and P-value distributions. The latter, in particular, is a guarantee that the estimated track parameters and errors associated are correctly computed.

The requirement of being able to measure a 10% change in emittance to 1% accuracy means that the transverse momentum resolution must be better than 10% of the beam RMS [10]. Fig. 5 shows the resolution of the track parameters. The ratio between the RMS in these distributions and the beam RMS is calculated and found to be well bellow the requirement. This gives a good indication from MC that MICE can measure the cooling effect to the required precision.

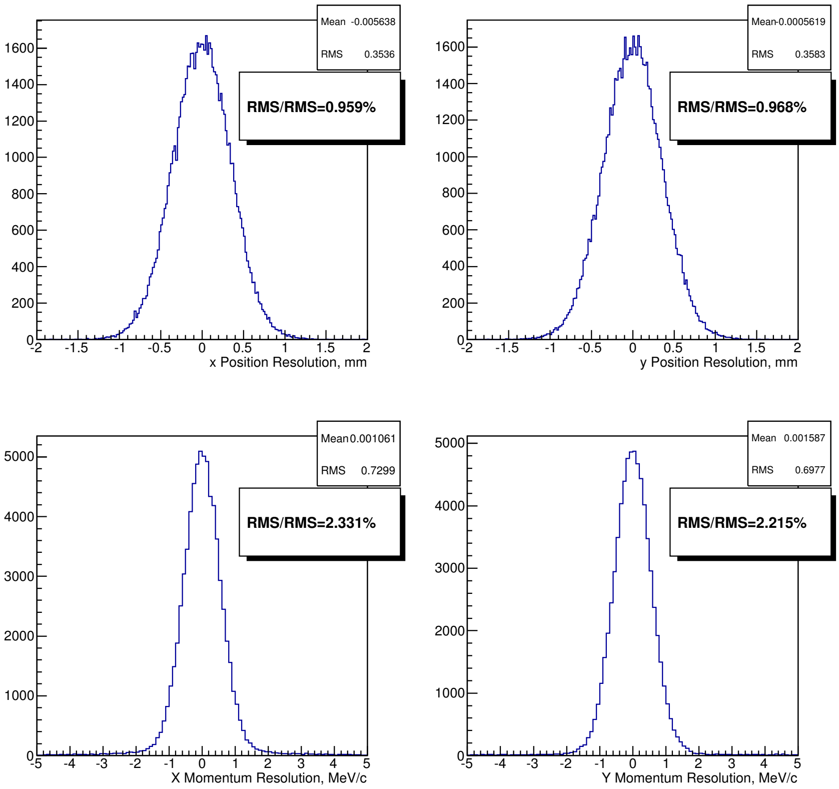

Figure 5. Resolution of the track parameters computed as the difference between MC truth and reconstruction values. The distribution RMS to beam RMS ratio is also shown.

References

[1] Skrinsky A N 1971, *Intersecting storage rings at Novosibirsk, Proceedings of Morges Seminar 1971*

[2] Bogomilov M *et al.* 2012 *JINST* **7** 5009

[3] Ellis M *et al.* 2011 *Nucl. Instrum. Meth.* A **659** 136

[4] Tunnell C D and Rogers C T 2011 *Proc. 2nd Int. Conf. Acc. Phys. (IPAC11, San Sebastian)* p 850

[5] Agostinelli S *et al.* 2003 *Nucl. Instrum. Meth.* A **506** 250

[6] Brun R and Rademakers F 1997 *Nucl. Instrum. Meth.* A **389** p 81 See also http://root.cern.ch/

[7] Frodensen A G, Skjeggestad O and Tofte H 1979 *Probability and Statistics in Particle Physics* (Bergen, Norway: Universitetsforlaget) p 259

[8] Kalman R E 1960 *Transactions of the ASME–Journal of Basic Engineering* **82** p 35

[9] Fruhwirth R 1987 *Nucl. Instrum. Meth.* A **262** p 444

[10] Khan A *et al.* 2005 *MICE Scintillating Fibre Tracker 1st Progress Report* MICE Note **90** (http://mice.iit.edu)